```
SELECT * FROM data_cleaning_project.laptop;

# Creating backup: 1st step

# 1.step is create the table structure
create table laptop_backup like laptop;
# 2. Step Insert all the value into backup table
insert into laptop_backup
select * from laptop;

# step 2: cheking how many rows and columns are there
select count(*) from laptop;

# step 3: cheking how much memory does the data occuipy;
select * from information_schema.tables
where TABLE_SCHEMA = 'data_cleaning_project' and
table_name = 'laptop';

# 278528 bytes in order to convert bytes into kb divid it 1024
select data_length/1024 from information_schema.tables
where TABLE_SCHEMA = 'data_cleaning_project' and
table_name = 'laptop';

# step 4 Drop non important columns
# unnamed: 0 columns is unrelevent so drop
select * from laptop;
Alter table laptop drop column `Unnamed: 0`;

# Step 5 Drop null values

with ind as (select * from laptop
where company is null and TypeName is null and Inches is null
and ScreenResolution is null and Cpu is null and Ram is null
and Memory is null and Gpu is NULL and OpSys is null
and Weight is null and Price is null)

# so now i have to find all those index that have null value
delete from laptop where `index` in (select `index` from ind);


# Step 6 drop duplicate

select count(*) from laptop;
```

```
with duplicates as(select Company,TypeName,Inches,ScreenResolution
,cpu,ram,Memory,gpu,opsys,Weight,Price,min(`index`) as duplicate_index
 from laptop group by Company,TypeName,Inches,ScreenResolution
,cpu,ram,Memory,gpu,opsys,Weight,Price)

DELETE from laptop where `index` not in (select duplicate_index from duplicates)
;
select * from laptop;

# Step 7 cleaning coumns:

select distinct(Company) from laptop; # This column is perfect fine no null value are there

select distinct(TypeName) from laptop; # This column is perfect no null value are there

# Inches datatype is text data i need to convert it into integer or double;
Alter table laptop modify column Inches decimal(10,1);
select * from laptop;

# Removing gb word from ram column  8Gb,16Gb....

UPDATE laptop t1
SET Ram = (
   SELECT replace(t2.ram, 'GB', '')
   FROM (SELECT * FROM laptop) t2
   WHERE t2.index = t1.index
);
# Converting text column into integer
alter table laptop modify column ram integer;

# cleaning on memory
update laptop t1
set memory = '64GB'
where t1.index = 720;

# Working on weight column
update laptop t1
set weight = '2.5kg'
where t1.index = (select `index` from (select * from laptop) t2 where weight = '?');
# Replace kg with ''
update laptop t1
set weight =
(select replace(weight,'kg','') from (select * from laptop) t2 where t2.`index` = t1.index);
```

```
# round the avg price and set the value
update laptop t1
set price = (
select round(price) from (select * from laptop) t2 where t1.index = t2.index);

select * from laptop;

# Modifying the price column into integer for memory efficency
ALTER table laptop modify column price integer;

# Now workng with difficult column;
# working with operating system
-- mac
-- windows
-- linux
-- no os
-- android chrome(others)
select opsys,
case
        when opsys like '%mac%' then 'macos'
   when opsys like '%windows%' then 'windows'
   when opsys like '%linux%' then 'linux'
   when opsys like '%No OS%' then 'NA'
   else 'others'
end as osbrand
from laptop;

update laptop
set opsys = case
        when opsys like '%mac%' then 'macos'
   when opsys like '%windows%' then 'windows'
   when opsys like '%linux%' then 'linux'
   when opsys like '%No OS%' then 'NA'
   else 'others'
end;

select * from laptop;

alter table laptop
add column gpu_brand varchar(255) after gpu,
add column gpu_name varchar(255) after opsys ;

update laptop t1
```

```
set gpu_brand = (
select substring_index(gpu,' ',1) as gp_brand from (select * from laptop) t2 where t1.index =
t2.index);


update laptop t1
set gpu_name = (
select replace(gpu,gpu_brand,'') as gp_name from (select * from laptop) t2 where t1.index =
t2.index);


alter table laptop drop column Gpu;

## Working with memory column
## I have to crate two new column that contain how much memory
## and wheathe it is ssd or othere stroage

alter table laptop
add column memory_space varchar(255) after memory,
add column memory_type varchar(255) after memory_space;

# sperating the value '128GB SSD +  1TB HDD' into '128GB SSD' and  '1TB HDD' and get first
2 value '128GB SSD' and put it into memory space
update laptop t1
set memory_space = (select substring_index(memory,' ',2) from (select * from laptop) as t2
where t1.index=t2.index);

# replace the value '128GB SSD +  1TB HDD' into '' and  '1TB HDD' and get first 2 value '1TB
HDD' and put it into memory type
update laptop t1
set memory_type = (select replace(memory,memory_space,'') from (select * from laptop) as t2
where t1.index=t2.index);

## Creating anothere column to put some important information
alter table laptop
add column extend_memory varchar(255) after memory_type;

update laptop t1
set extend_memory = (
select
case
        when memory_type like '%TB%' then substring_index(memory_type,' +  ',-1)
   else NULL
end as ext
```

```
from (select * from laptop) t2 where t1.index = t2.index);

SELECT * from laptop;
## SPerating the memory type value if the value is like 250GB + 1TB HDD
update laptop t1
set memory_type = (
select
case
        when memory_type like '%+%TB%' then replace(memory_type,extend_memory,'')
   else memory_type
end as new_me_type
from (select * from laptop) t2 where t1.index = t2.index);

# Replce the memory type value if memory has SDD + 1TB with SDD 1TB and update the value

update laptop t1
set memory_type = (
select replace(memory_type,'+','') from (select * from laptop) t2 where t1.index= t2.index);


# update the memory type column if a column has a value like ' SSD   SDD' into SDD

update laptop t1
set memory_type = (
select
 case
        when memory_type = ' SSD    SSD' then 'SSD'
   else memory_type
 end as 'new_memory_type'
 from (select * from laptop) t2 where t1.index = t2.index);

## Creating a column for extend_memory_type
alter table laptop
add column extend_memory_type varchar(255);

## Separating extend_memory 1TB HDD to 1TB and HDD and updating the value to set into
extend_memory_type
update laptop t1
set extend_memory_type =
(select substring_index(extend_memory,' ',-1) from (select * from laptop) t2 where t1.index =
t2.index);

## processing extend_memory 1TB HDD to 1TB
update laptop t1
```

```
set extend_memory=(
select replace(extend_memory,' HDD','') from (select * from laptop) t2 where t1.index =
t2.index);

select * from laptop;
```